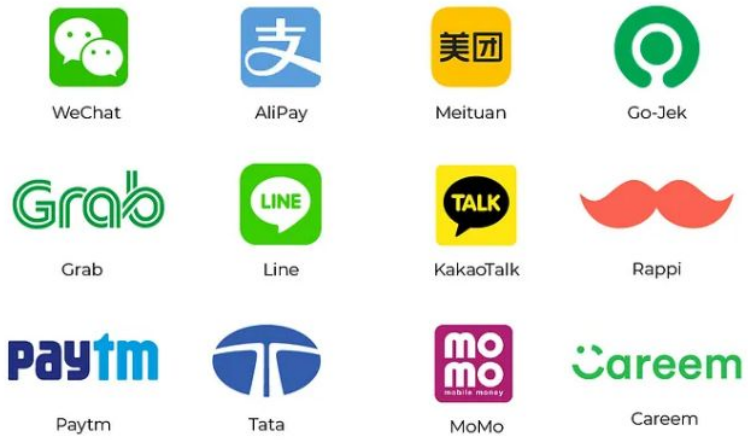
MiniCAT: Understanding and Detecting Cross-Page Request Forgery Vulnerabilities in Mini-Programs

Zidong Zhang, Qingsheng Hou, Lingyun Ying, Wenrui Diao, Yacong Gu, Rui Li, Shanqing Guo, Haixin Duan

Oct 15th, 2024

Background: Mini-program

- **Mini-program: A new era of mobile apps...**
  - ▯**Lightweight**: No Download
  - ▯**Global**: WeChat, Baidu, TikTok, Alipay, LINE...
  - ▯**Popularity**: 900+ million Users
  - ▯**Mutli Scenario**: E-shop, Orders, Taxis..



Elon Musk: "...It's sort of like Twitter, plus PayPal, plus a whole bunch of other things. And all rolled into one great interface."

The Arch of Mini-programs: WeChat Case

- **Front-End:**
  - **Render Layer:** WXML + WXSS
  - **Logic Layer:** JavaScript-based
- **Back-End: with Super App**

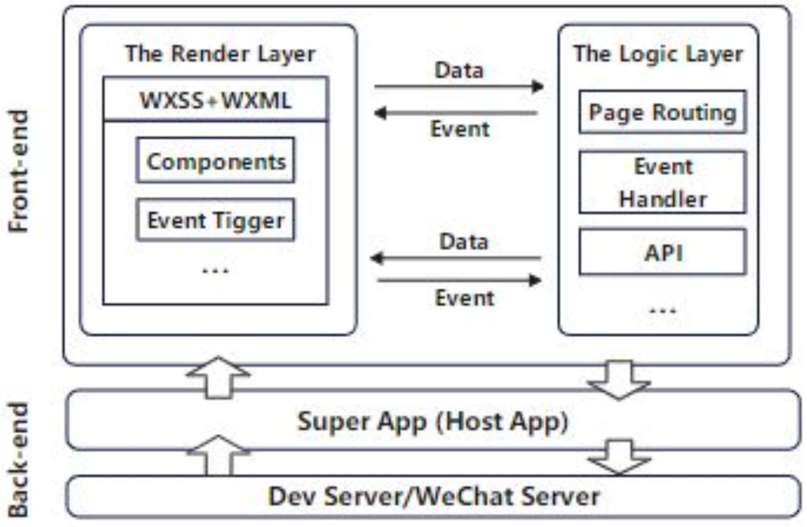


Figure 1: WeChat mini-program architecture.

- **Mobile-apps-like & Web-apps-like**
  - An App (≈Mobile App) in a Super App (≈OS).
  - A **Web application** in the framework (≈Browser).

Mini-program Page Routing

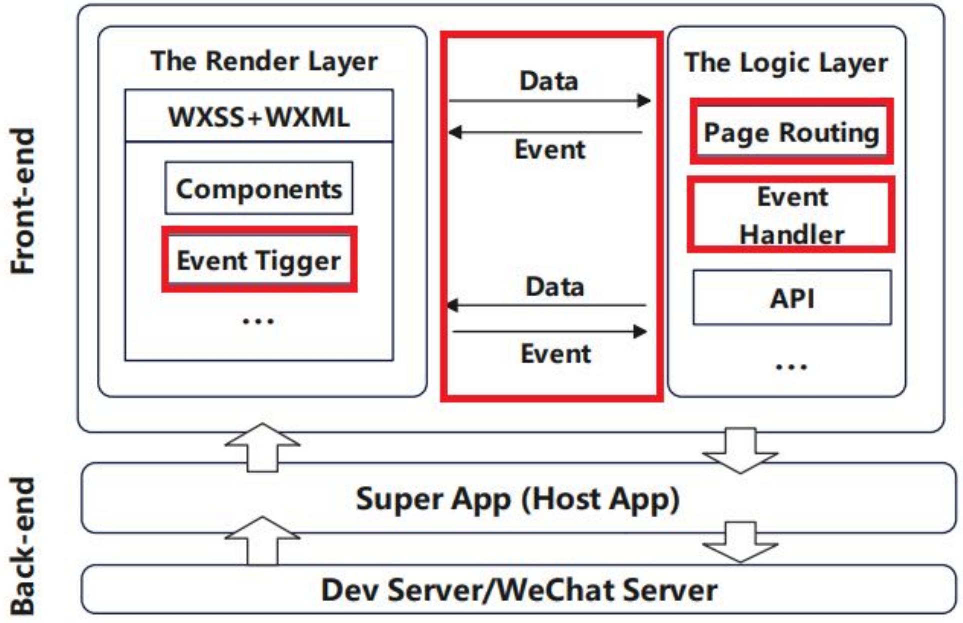
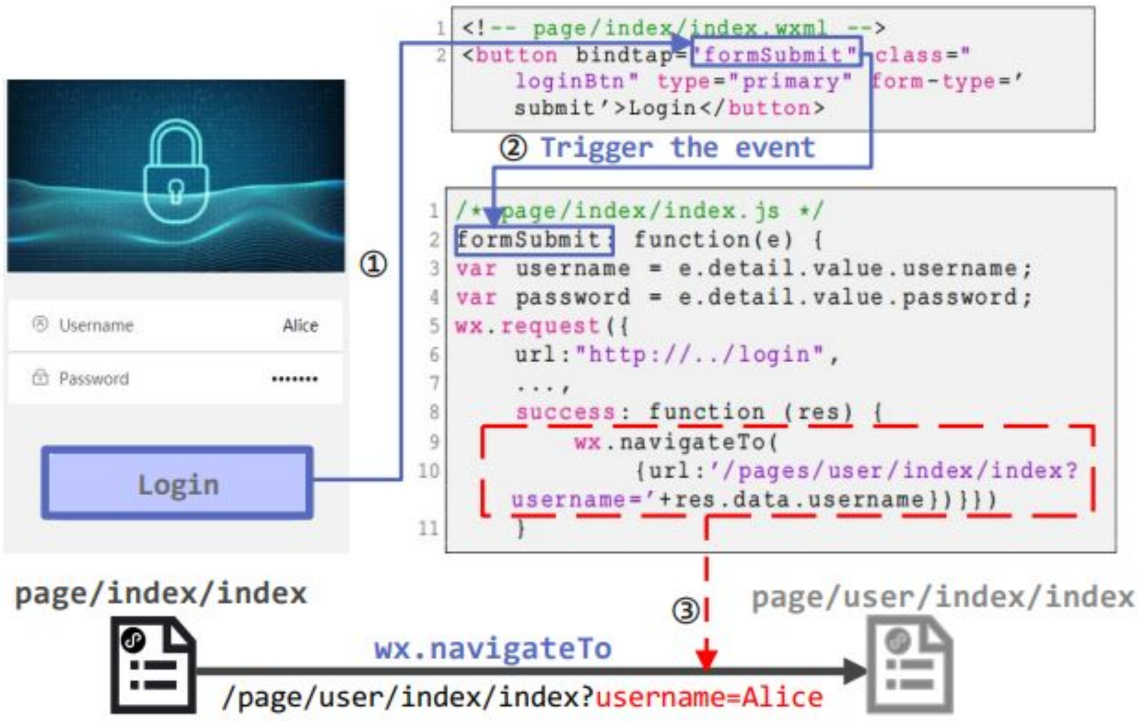


Figure 1: WeChat mini-program architecture.



```
<!-- page/index/index.wxml -->
<button bindtap="formSubmit" class="
    loginBtn" type="primary" form-type='
    submit'>Login</button>
```

```
/* page/index/index.js */
formSubmit: function(e) {
var username = e.detail.value.username;
var password = e.detail.value.password;
wx.request({
    url:"http://../login",
    ...,
    success: function (res) {
        wx.navigateTo(
            {url:'/pages/user/index/index?
    username='+res.data.username})}}})
    }
```

- **Page routing APIs**: wx.navigateTo, wx.reLaunch, wx.redirectTo
- **Transparent**: not visible.

Mini-program Page Routing V.S. Web Routing

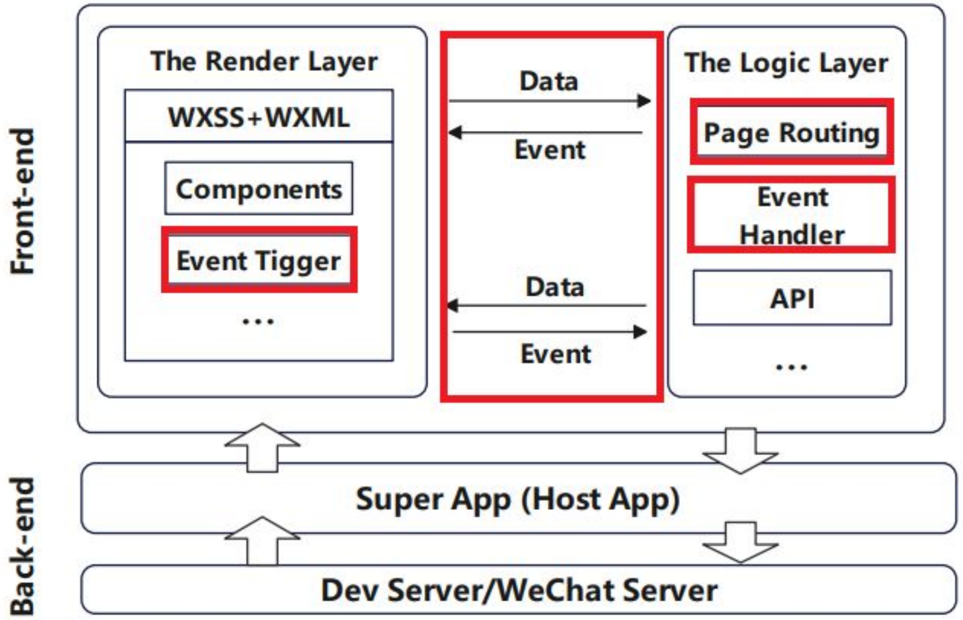
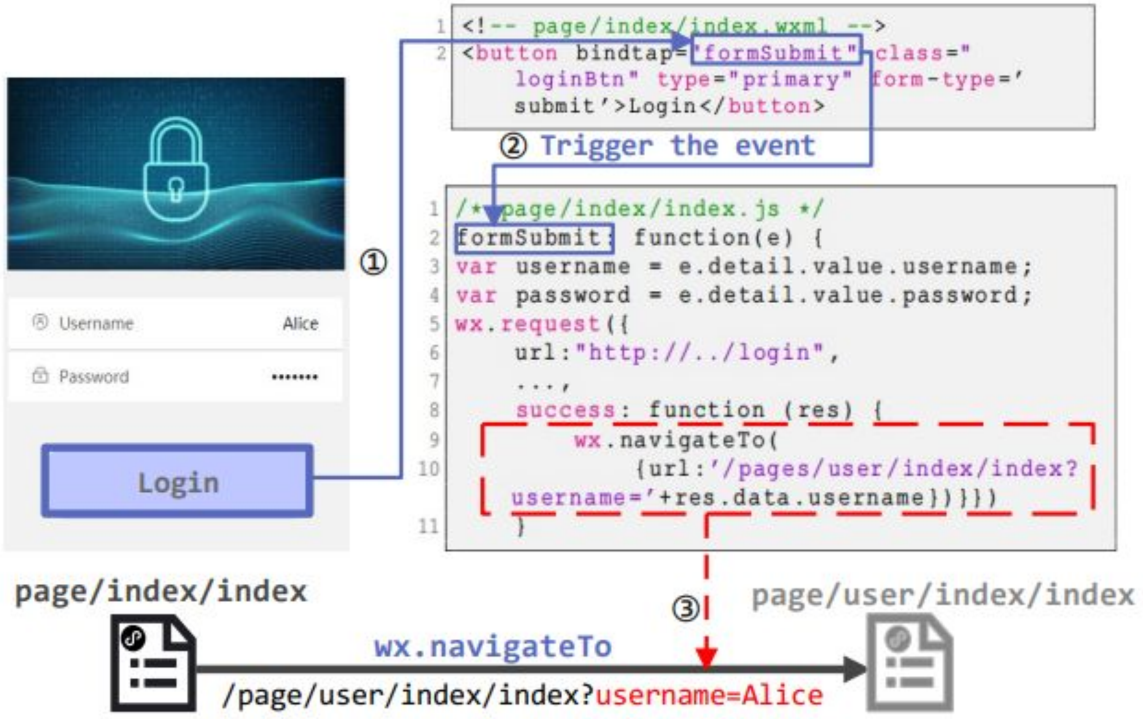


Figure 1: WeChat mini-program architecture.



```
<!-- page/index/index.wxml -->
<button bindtap="formSubmit" class="
    loginBtn" type="primary" form-type='
    submit'>Login</button>
```

```
/* page/index/index.js */
formSubmit: function(e) {
var username = e.detail.value.username;
var password = e.detail.value.password;
wx.request({
    url:"http://../login",
    ...,
    success: function (res) {
        wx.navigateTo(
            {url:'/pages/user/index/index?
    username='+res.data.username})}})
    }
```

/page/user/index/index?username=Alice

- Passing parameters **via URL Schema**: HTTP-GET Method Like
- **ONLY support Plaintext transmission.**

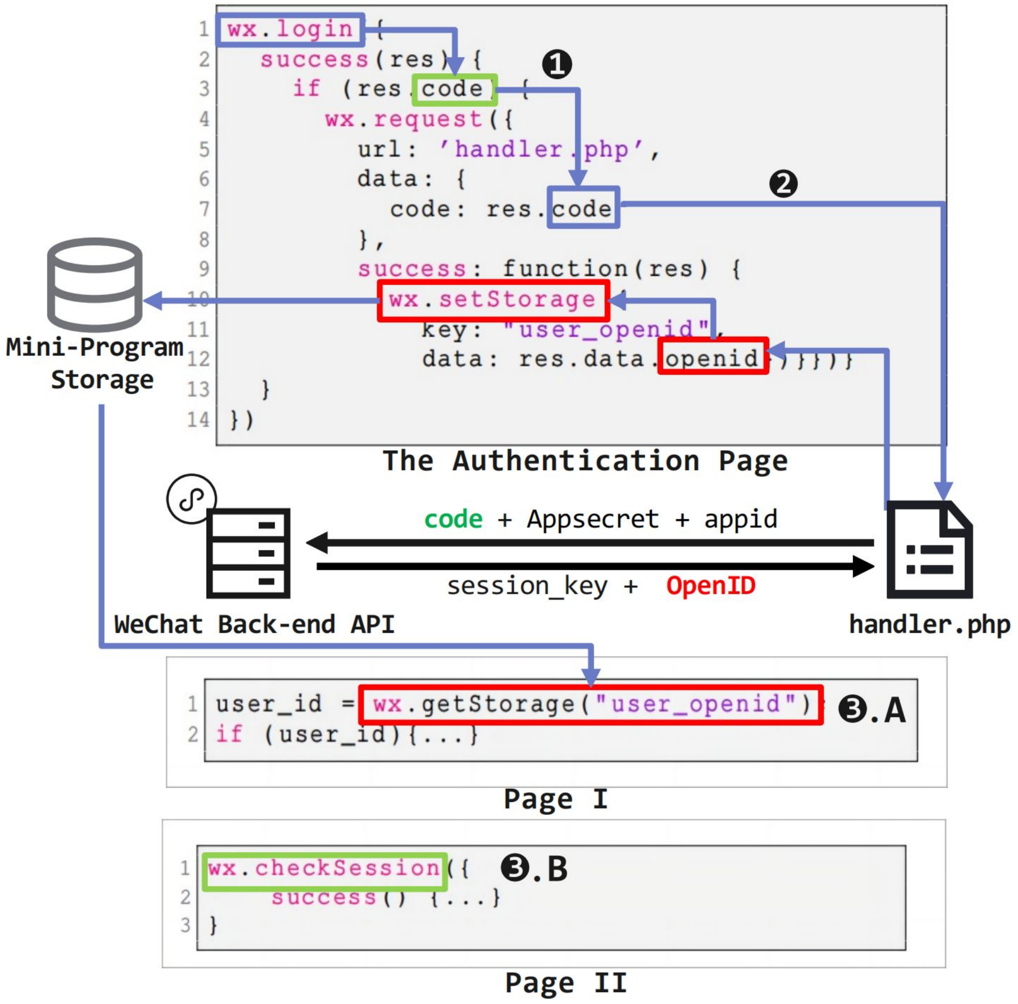
Mini-program User State

- User State ≈ Cookie & Session in HTTP
- Two method to check:

①: **wx.login → code → session_key**
□**Check**: `wx.checkSession()`

②: **wx.login → code → OpenID**
□**Check**: Customize via storage
`wx.setStorage ↔ wx.getStorage`

Need to verify on every page :(



```
wx.login({
  success(res) {
    if (res.code) {
      wx.request({
        url: 'handler.php',
        data: {
          code: res.code
        },
        success: function(res) {
          wx.setStorage({
            key: "user_openid",
            data: res.data.openid})}})}
  }
})
```

The Authentication Page

```
user_id = wx.getStorage("user_openid");
if (user_id){...}
```

Page I

```
wx.checkSession({
    success() {...}
}
```

Page II

New Vuln : Cross-Page Request Forgery (MiniCPRF)

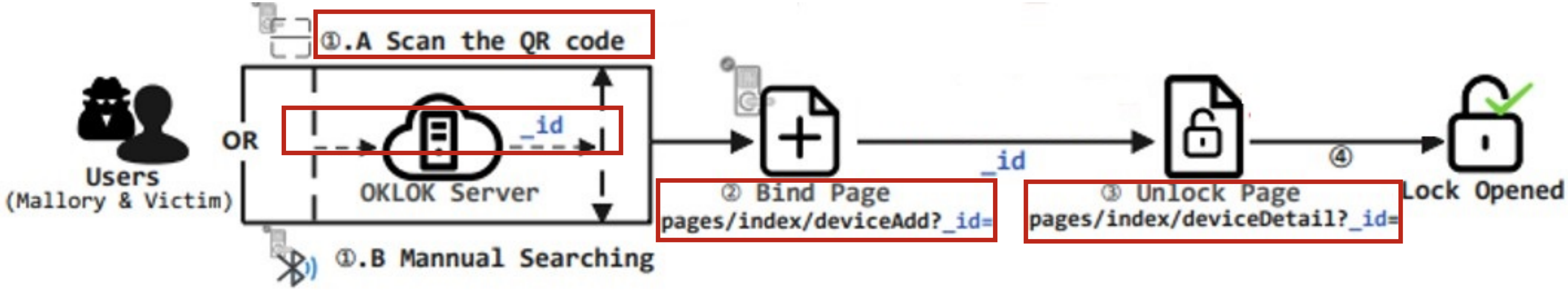
- **Page Routing:** `page/index/login?pwd=xxxxx`
  - Plaintext transimission
  - Parameters conveyed by URL Schema

- **User State: NOT** support Cookie-like features
  - Custom User State: Need to verify on every page

How to modify page routing URLs in mini-programs?

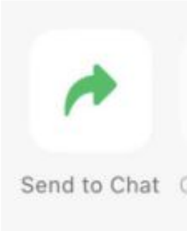
A Motivation Case: Unauthorized Unlocking

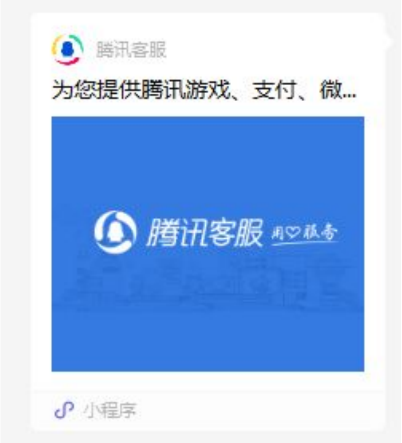
I. The Normal Path



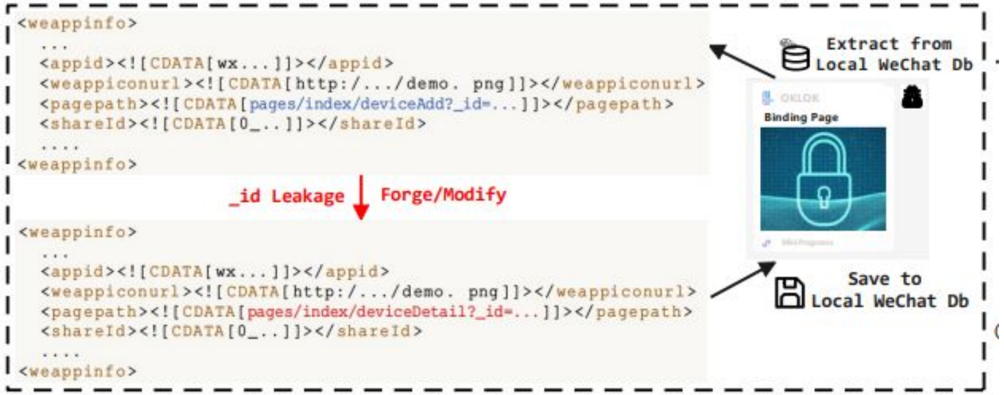
Sharing and Forwarding of Mini-programs

- **Sharing** : Generate **a mini-program card**
- **The Mini-program Card:**
  - A XML Format text in the local Db
  - .. And This Db can be decrypted & modified !





Bingo! The card can be modified ;)



```
<weappinfo>
  ...
  <appid><![CDATA[wx...]]></appid>
  <weappiconurl><![CDATA[http:/.../demo. png]]></weappiconurl>
  <pagepath><![CDATA[pages/index/deviceAdd?_id=...]]></pagepath>
  <shareId><![CDATA[0_..]]></shareId>
  ....
<weappinfo>
```

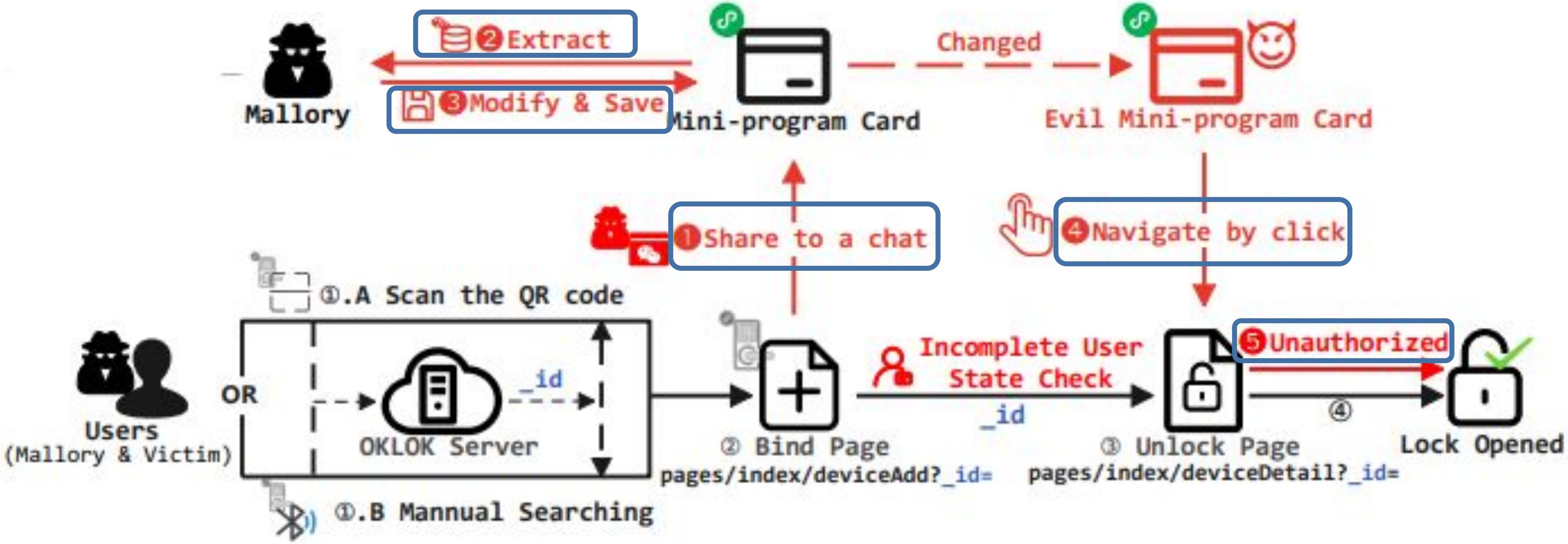
_id Leakage ↓ Forge/Modify

```
<weappinfo>
  ...
  <appid><![CDATA[wx...]]></appid>
  <weappiconurl><![CDATA[http:/.../demo. png]]></weappiconurl>
  <pagepath><![CDATA[pages/index/deviceDetail?_id=...]]></pagepath>
  <shareId><![CDATA[0_..]]></shareId>
  ....
<weappinfo>
```

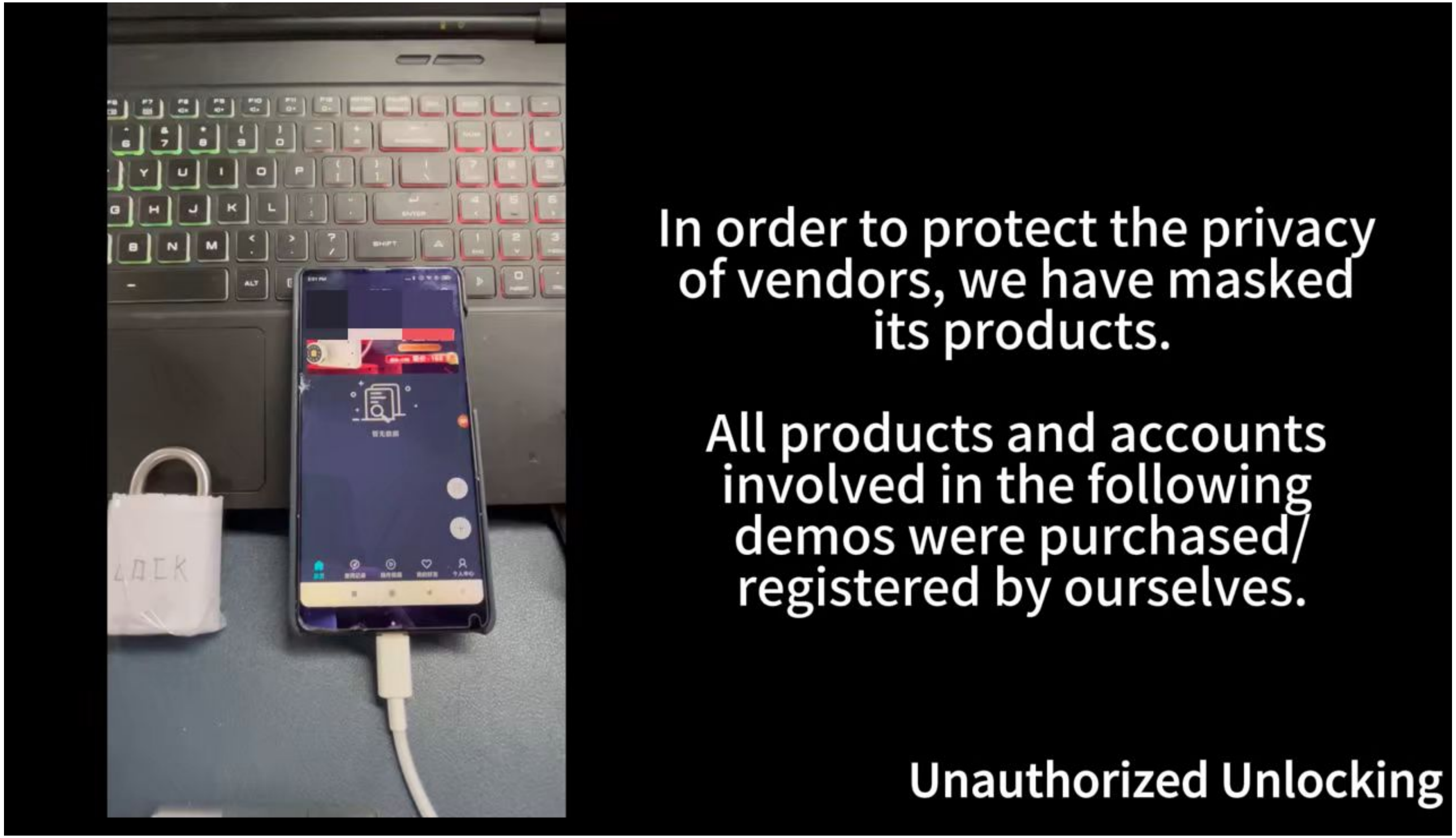
A Motivation Case: Unauthorized Unlocking

II. The Attack Path

- The Bind Page **can be shared.**
- And Unlock Page: **Incomplete User State check**



Demo Video



In order to protect the privacy of vendors, we have masked its products.

All products and accounts involved in the following demos were purchased/ registered by ourselves.

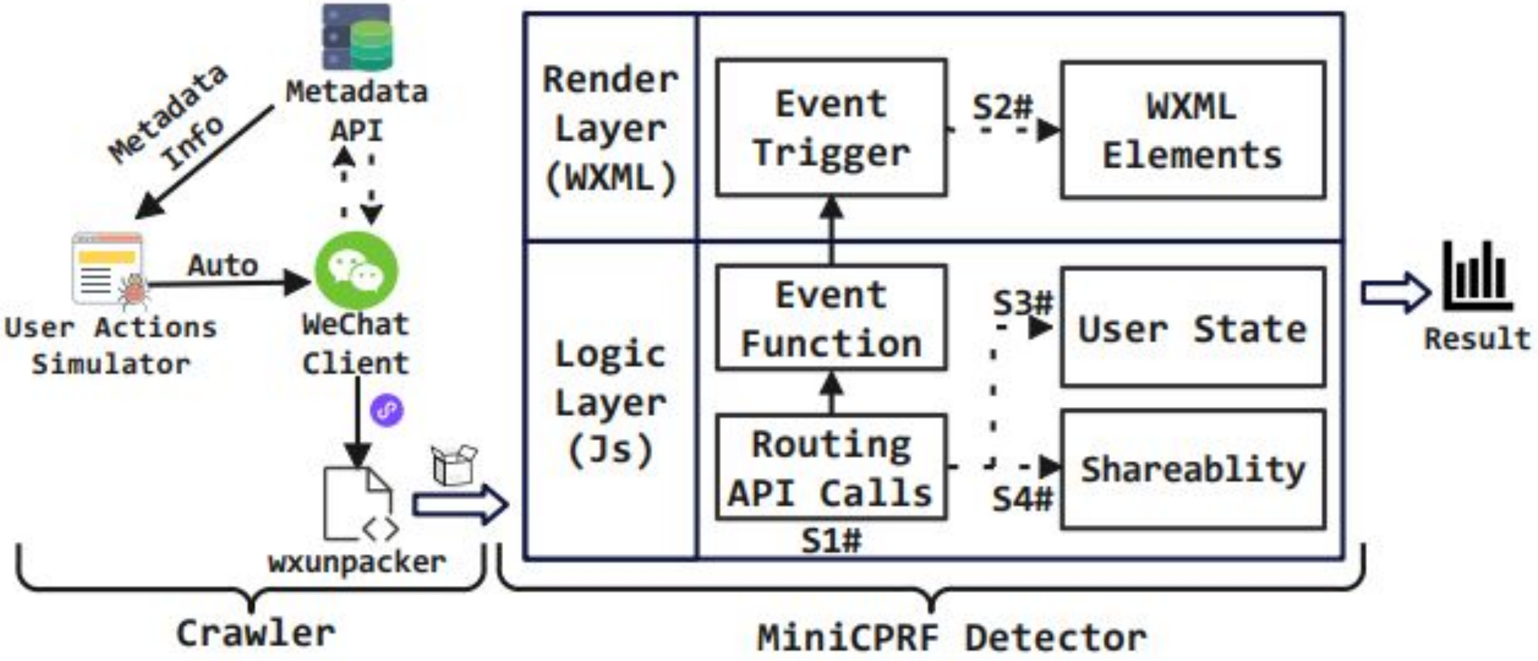
Unauthorized Unlocking

How to Automate Analysis?

- Three Steps for MiniCPRF:
- S1: Identify vulnerabilities and extract URL parameters.
  - Q1: Where are routing APIs: wx.navigateTo, wx.redirecTo,wx.reLaunch?
- S2: Modify or create mini-program card with modified URL.
  - Q2: How to get to the vuln page?
  - Q3: Does the vuln page implement a complete user state check?
- S3: Click the modified/generated card.
  - Q4: Can the trigger page be shared to generate the card?

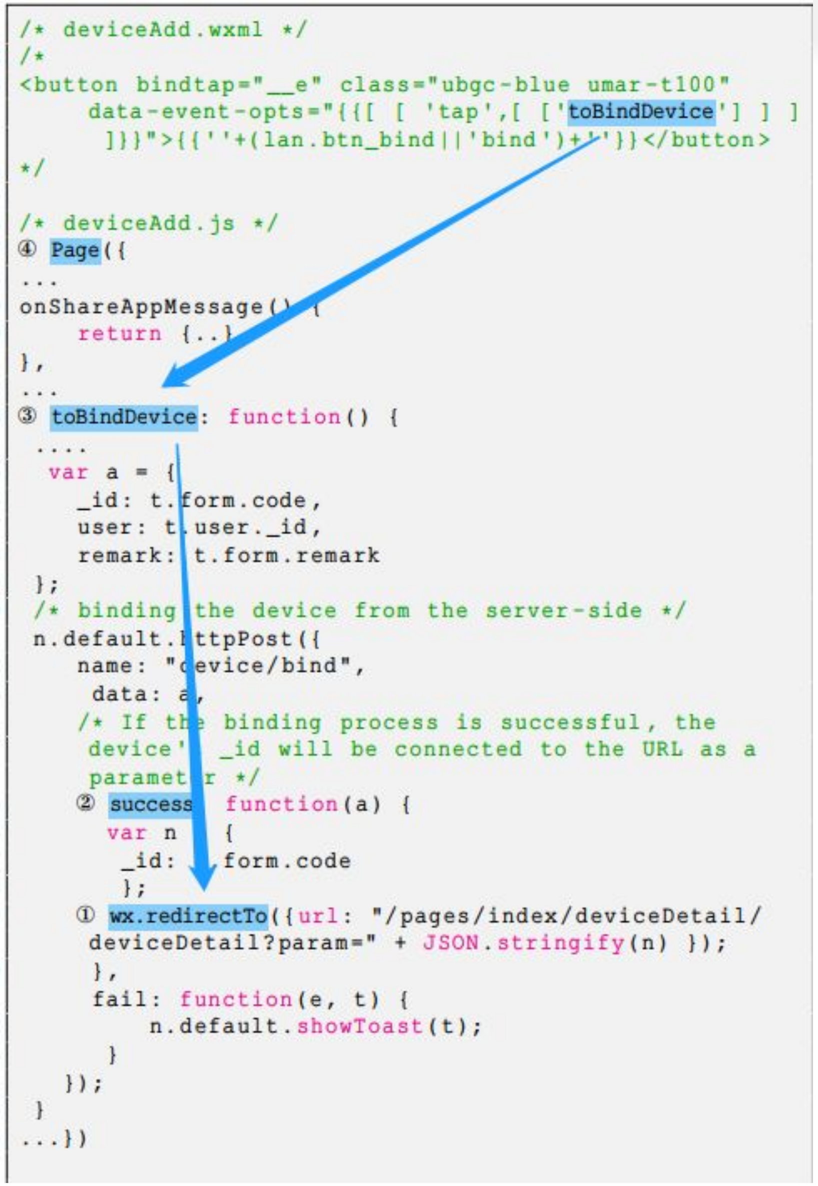
Our Solution: MiniCAT

- A Mini-program Crawler & A MiniCPRF Detector.
- Crawler: Stimulate User-action, Collected 44k+.



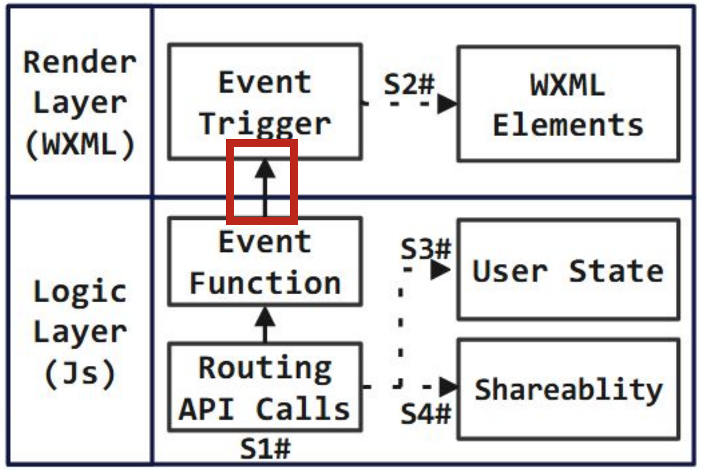
Our Solution: MiniCAT

- A MiniCPRF Detector
  - Q1: Where are routing API?
    - Building AST & Found Callee Nodes
  - Q2: How to get to the vuln page?
    - Convert WXML to HTML
    - Reverse Taint：Logic layer → Render Layer



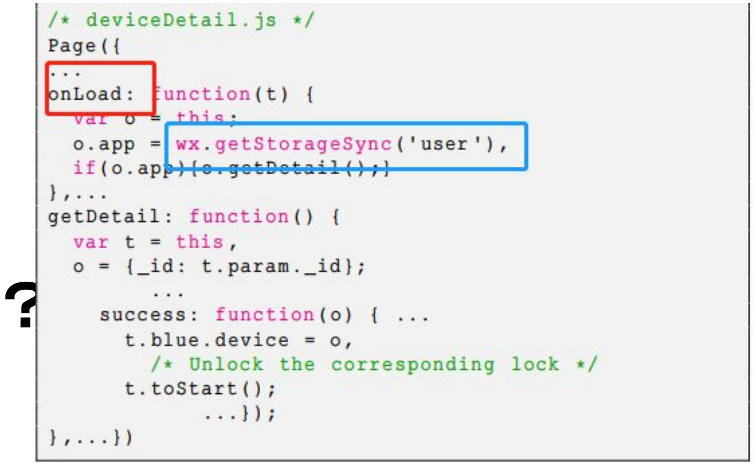
```
/* deviceAdd.wxml */
/*
<button bindtap="__e" class="ubgc-blue umar-t100"
     data-event-opts="{{[ [ 'tap',[ ['toBindDevice'] ] ]
      ]}}">{{''+(lan.btn_bind||'bind')+''}}</button>
*/

/* deviceAdd.js */
④ Page({
...
onShareAppMessage() {
    return {..}
},
...
③ toBindDevice: function() {
 ....
  var a = {
    _id: t.form.code,
    user: t.user._id,
    remark: t.form.remark
 };
 /* binding the device from the server-side */
 n.default.httpPost({
    name: "device/bind",
     data: a,
    /* If the binding process is successful, the
     device's _id will be connected to the URL as a
     parameter */
    ② success: function(a) {
      var n = {
       _id: t.form.code
       };
    ① wx.redirectTo({url: "/pages/index/deviceDetail/
     deviceDetail?param=" + JSON.stringify(n) });
     },
     fail: function(e, t) {
         n.default.showToast(t);
      }
   });
 }
...})
```

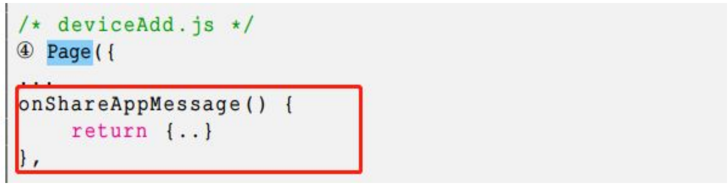


Our Solution: MiniCAT

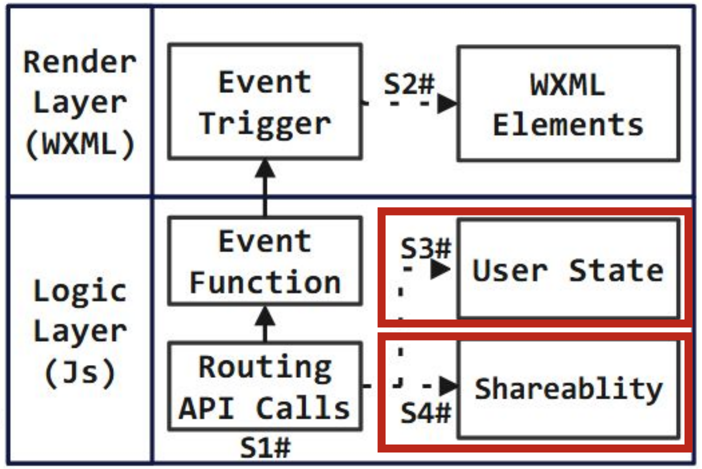
- **A MiniCPRF Detector**
  - **Q3: The integrity of user state check?**
    - Check the page loading function
  - **Q4: Can the page be shared to generate the card？**
    - Shareable control API



```
/* deviceDetail.js */
Page({
...
onLoad: function(t) {
  var o = this;
  o.app = wx.getStorageSync('user'),
  if(o.app){o.getDetail();}
},...
getDetail: function() {
  var t = this,
  o = {_id: t.param._id};
        ...
    success: function(o) { ...
      t.blue.device = o,
        /* Unlock the corresponding lock */
      t.toStart();
            ...});
},...})
```



```
/* deviceAdd.js */
④ Page({
...
onShareAppMessage() {
    return {..}
},
```



Measurement Result

- **Analyzed 41,726/44,273 (94.2%)**
  - **13,349/41,726 (32.0%)** as potentially vulnerable
- **Random selected 400 Mini-programs：to verify**
  - **316/400 (79.0%)** confirmed. 3 CNVDs
  - Fp: 38/400 (9.5%)，Fn: no benchmark
- **Insight Measurement**
  - Based on Categories
  - Template Mini-programs
  - Passive DNS: by domain whitelist → Popularity

One-Page Take Away

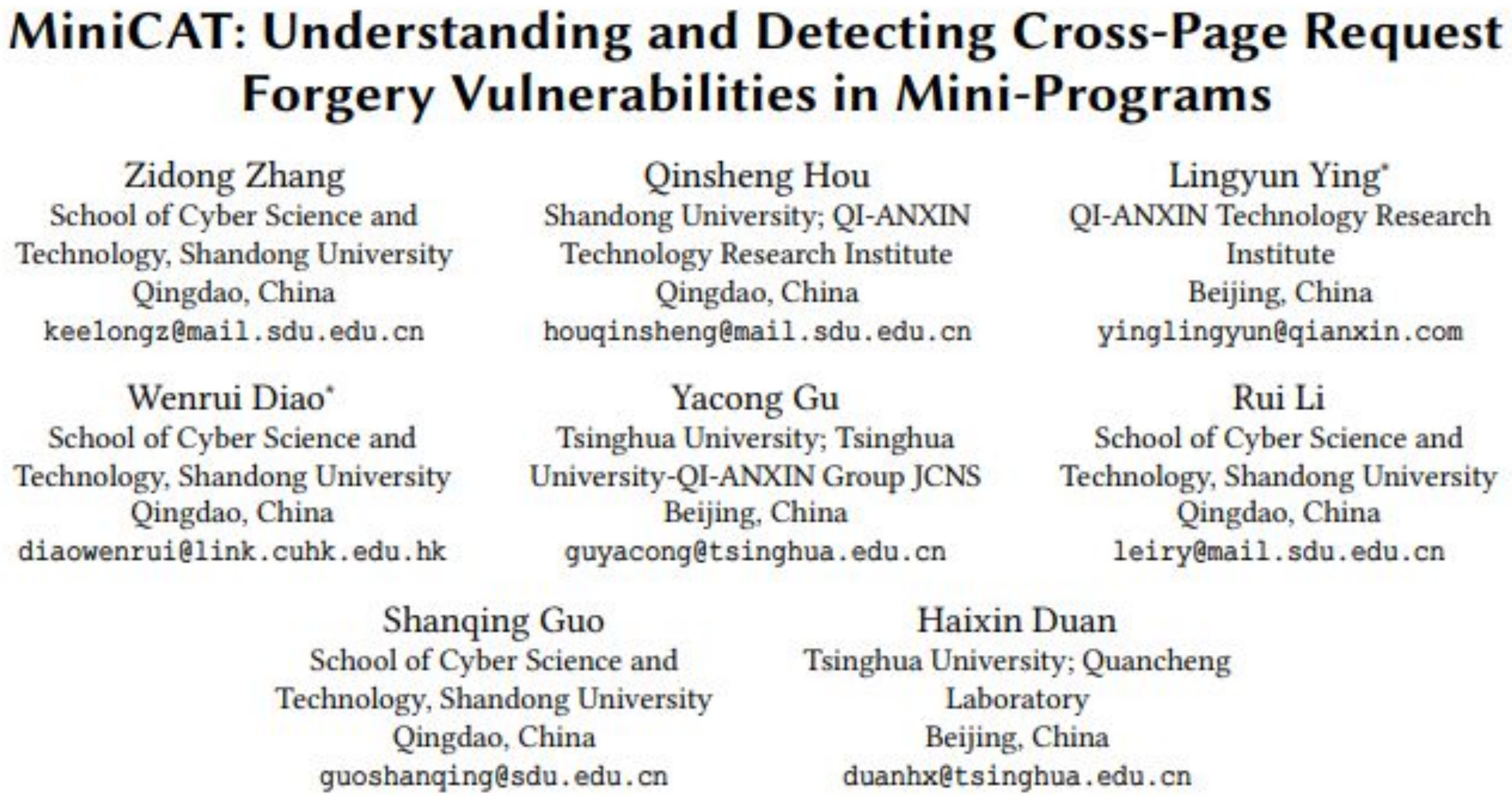
- Vulnerability Discovery :MiniCPRF
  - A new kind of vulnerability in mini-programs: Cross Page Request Forgery

- Vulnerability Detection: MiniCAT
  - A automatic detector based on code analysis.
  - Identify a series of risks in the real-world evaluation.

- Open Access:
  - MiniCAT: https://github.com/kee1ongz/MiniCAT
  - Attack Demo Site: https://sites.google.com/view/minicprf
  - Contact the author: kee1ongzz@gmail.com

Q&A

Thanks!



MiniCAT: Understanding and Detecting Cross-Page Request Forgery Vulnerabilities in Mini-Programs

Zidong Zhang
School of Cyber Science and Technology, Shandong University
Qingdao, China
keelongz@mail.sdu.edu.cn

Qinsheng Hou
Shandong University; QI-ANXIN Technology Research Institute
Qingdao, China
houqinsheng@mail.sdu.edu.cn

Lingyun Ying*
QI-ANXIN Technology Research Institute
Beijing, China
yinglingyun@qianxin.com

Wenrui Diao*
School of Cyber Science and Technology, Shandong University
Qingdao, China
diaowenrui@link.cuhk.edu.hk

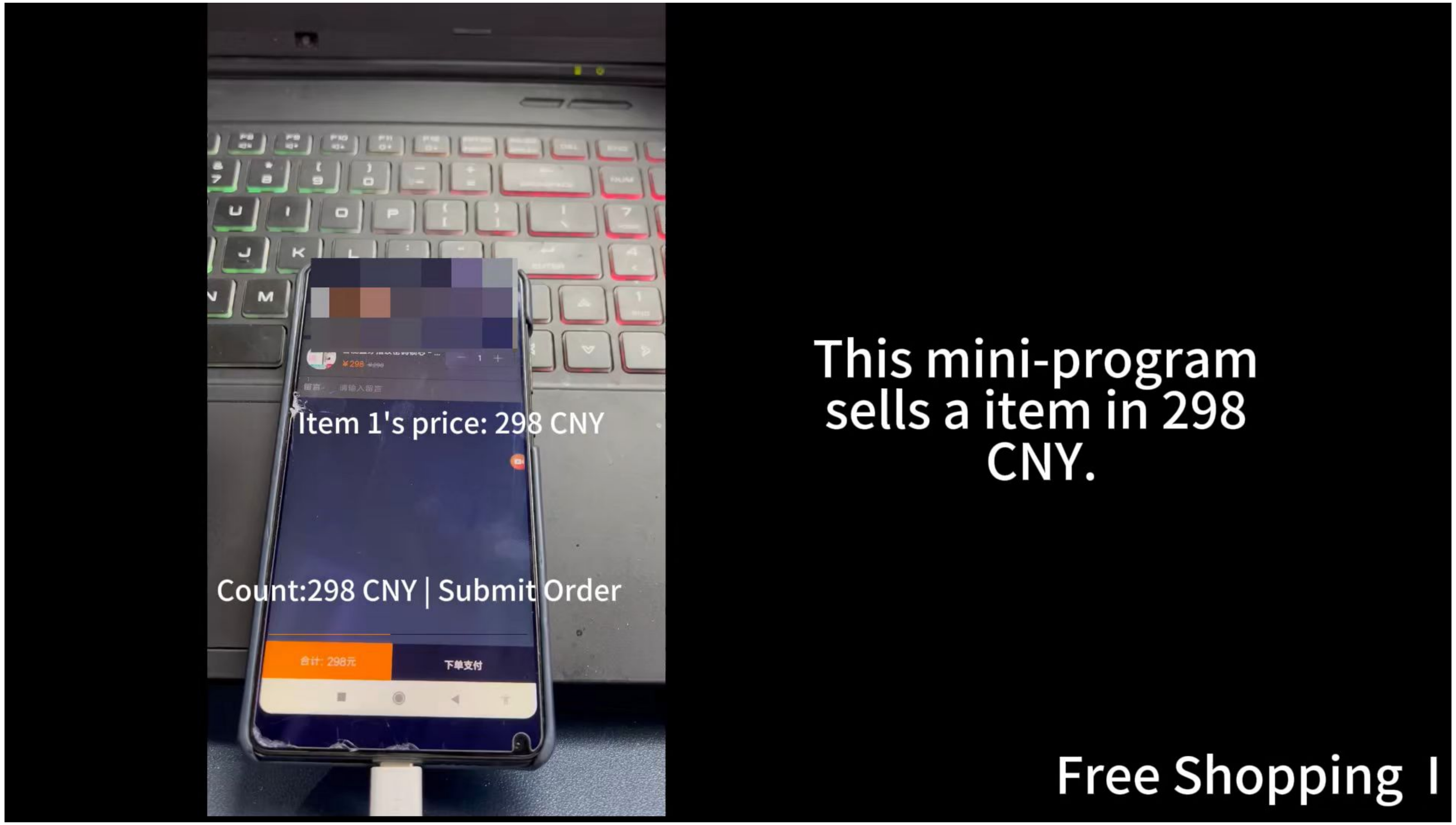
Yacong Gu
Tsinghua University; Tsinghua University-QI-ANXIN Group JCNS
Beijing, China
guyacong@tsinghua.edu.cn

Rui Li
School of Cyber Science and Technology, Shandong University
Qingdao, China
leiry@mail.sdu.edu.cn

Shanqing Guo
School of Cyber Science and Technology, Shandong University
Qingdao, China
guoshanqing@sdu.edu.cn

Haixin Duan
Tsinghua University; Quancheng Laboratory
Beijing, China
duanhx@tsinghua.edu.cn

Demo: Free Shopping



This mini-program sells a item in 298 CNY.

Free Shopping I

Measurements on Multi Platforms

- **Similar Mechanism → Similar Vulnerability**
  - Verified in same-name mini-programs.
  - **MiniCAT: Support ALL :)**

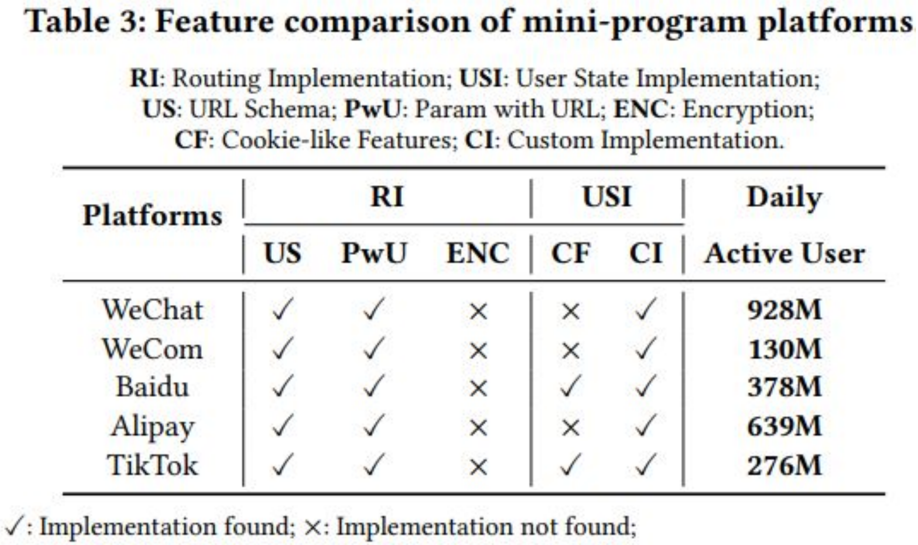
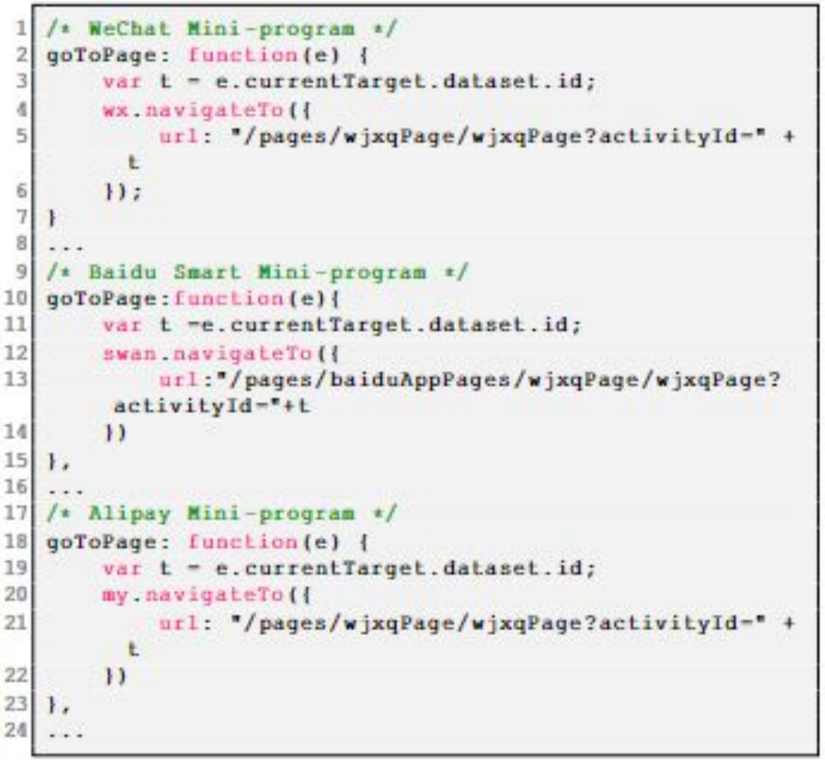


Table 3: Feature comparison of mini-program platforms

RI: Routing Implementation; **USI**: User State Implementation;
US: URL Schema; **PwU**: Param with URL; **ENC**: Encryption;
CF: Cookie-like Features; **CI**: Custom Implementation.

Platforms	RI			USI		Daily
	US	PwU	ENC	CF	CI	Active User
WeChat	✓	✓	×	×	✓	928M
WeCom	✓	✓	×	×	✓	130M
Baidu	✓	✓	×	✓	✓	378M
Alipay	✓	✓	×	×	✓	639M
TikTok	✓	✓	×	✓	✓	276M

✓: Implementation found; ×: Implementation not found;



```
/* WeChat Mini-program */
goToPage: function(e) {
    var t = e.currentTarget.dataset.id;
    wx.navigateTo({
        url: "/pages/wjxqPage/wjxqPage?activityId=" +
          t
    });
}
...
/* Baidu Smart Mini-program */
goToPage:function(e){
    var t =e.currentTarget.dataset.id;
    swan.navigateTo({
        url:"/pages/baiduAppPages/wjxqPage/wjxqPage?
     activityId="+t
    })
},
...
/* Alipay Mini-program */
goToPage: function(e) {
    var t = e.currentTarget.dataset.id;
    my.navigateTo({
        url: "/pages/wjxqPage/wjxqPage?activityId=" +
          t
    })
},
...
```